*Узагальнені завадостійкі коди в задачах забезпечення цілісності інформаційних об'єктів в умовах природних впливів. К. НТУУ "КПІ" // Правове, нормативне та метрологічне забезпечення Системи захисту інформації в Україні. Випус*к 2 (13)// 2006, с. 144–159. **3.** *Акушский И. Я., Юдицкий Д. И. Машинная арифметика в остаточных классах. // М.: Сов. радио*, 1966. – 421 с. **4.** *Василенко В. С., Будько М. М., Короленко М. П. Контроль и відновлення цілісності інформації в автоматизованих системах. К. НТУУ "КПІ" // Правове, нормативне та метрологічне забезпечення Системи захисту інформації в Україні. Випус*к 4// 2002, с. 119–128.

УДК 004.658.2

АУДИТ АНОМАЛЬНОГО ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЕЙ БАЗ ДАННЫХ

Михаил Коломыцев, Светлана Носок
Национальный технический университет Украины «Киевский политехнический институту»

Анотації: **Вирішується задача розробки підходу до формування профілю типової поведінки, що враховує специфіку функціонування СУБД. Пропонується процедура виявлення аномальної поведінки на базі створеного профілю.**
Summary: **The task of developing an approach to typical behaviour profile that accounts for the database management system functioning pecularities is covered in his work. The procedure of anomalous behaviour detection on the basis of the developed profile is suggested.**
Ключові слова: **Аномальна поведінка, профіль користувача, байєсовський класифікатор.**

I Введение

Современные СУБД обладают развитыми возможностями обеспечения безопасности данных. Однако, в силу ряда причин, обусловленных сложностью информационных систем и изменчивостью обстановки их функционирования, становится возможным нарушение установленной политики безопасности. Зачастую [1], нарушителями политики безопасности становятся инсайдеры. Защита информации от инсайдеров является сложной задачей, поскольку они являются легитимными пользователями, действующими в рамках предоставленных им полномочий. В таких ситуациях особенное значение приобретает способность системы безопасности СУБД обнаруживать изменение в поведении пользователей, сигнализирующее о попытках злоумышленных действий. Преимуществом технологии обнаружения атак на сервер БД, основанной на обнаружении аномальной активности, в отличие от подхода с использованием сигнатур, является большая гибкость и возможность обнаруживать неизвестные атаки. В данной работе предлагается подход к обнаружению аномального поведения пользователей в базах данных, использующих ролевое управление доступом.

II Постановка задачи

Системы обнаружения аномального поведения основаны на том, что известны некоторые параметры, характеризующие правильное или допустимое поведение объекта наблюдения. В качестве таких параметров могут выступать, например, количественные показатели использования ресурсов сервера БД [2] или интенсивности обращений к ресурсам [5].Значения параметров, соответствующие нормальному поведению объекта наблюдения называется профилем. Выявления аномального поведения основано на сравнении текущих значений параметров активности с профилем. Параметры профиля вычисляются за достаточно большой период времени. Под текущими значениями параметров активности обычно понимаются значения, вычисленные на коротком интервале времени (применительно к СУБД – по одной или нескольким транзакциям), непосредственно предшествующем рассматриваемому моменту.

Хотя аномальное поведение не обязательно является следствием атаки на систему, с высокой долей вероятности оно свидетельствует о нарушении (умышленным или нет) политики безопасности. Поскольку действия злоумышленника обязательно чем-то отличаются от поведения обычного пользователя, в работах [3,4] методы обнаружения аномального поведения положены в основу систем обнаружения вторжений в базу данных.

Задача обнаружения аномального поведения пользователей баз данных имеет существенные особенности. Во-первых, то, что считается аномальным с точки зрения функционирования БД, вовсе не является таковым с точки зрения операционной системы или сетевых коммуникаций. Следовательно, созданные для них аналогичные сервисы попросту «не увидят» факта нарушения политики безопасности. Во-вторых, эти сервисы во многом ориентированы на защиту от атак извне, и в меньшей степени защищают от внутренних угроз.

В данной работе авторы решают задачу разработки подхода к формированию профиля типичного поведения, учитывающего специфику функционирования СУБД. Предлагается процедура обнаружения аномального поведения на основании созданного профиля.

III Выбор объекта наблюдения

В качестве объектов наблюдения авторы предлагают использовать не пользователей баз данных (количество которых может измеряться десятками и сотнями) а роли БД $\boldsymbol{R_j \in R\ j=1,\ldots,n}$.

Ролевое управление доступом широко используется в современных СУБД и является эффективным средством разграничения доступа. Роли создаются для решения задач, функционально связанных с точки зрения прикладной системы, а пользователи, включенные в эти роли, выполняют сходные производственные обязанности. В некоторых случаях разделение на роли, может возникать не на уровне СУБД, а на уровне клиентского программного обеспечение (хотя бы с помощью ограничений, накладываемых интерфейсом клиентского ПО).

Независимо от того, каким образом организовано ролевое управление доступом, действия пользователей, включенных в эту роль, имеют много общего. Построение профиля для роли позволяет снизить требования к ресурсам сервера БД, необходимым для обнаружения аномалий.

VI Выбор параметров, на основании которых создается профиль

Характеристиками, определяющими поведение пользователя БД, являются выполняемые действия (команды DML) и таблицы, над которыми эти действия выполняются. По этому, при каждом обращении пользователя к таблицам БД регистрируется следующий набор параметров: фиксируется роль $\boldsymbol{R_j}$, от имени которой выполняются действия, и структура вида:

$$\{< c >,< T >,< A >\} \qquad (1)$$

где **c** – выполняемая команда,

<T> – список таблиц, которыми оперирует команда **c**,

<A> – список атрибутов таблиц **<T>,** участвующих в команде.

В структуре (1) команда **c** обозначается номером в соответствии со списком:

1 - Select, 2 - Insert, 3 - Update, 4 - Delete.

Если в команде участвуют **n** таблиц, то **<T>** – вектор размерности **n**, содержащий номера таблиц задействованных в команде. В этом случае **<A>** – вектор, состоящий из **n** векторов, причем ***j*-**ый вектор содержит список номеров атрибутов ***j*-**ой таблицы, над которыми выполняется команда. Нумерация таблиц и атрибутов в них произвольная, необходимо только обеспечивать ее стабильность при модификации структуры базы.

Команда Delete рассматривается как вырожденный случай (1) и регистрируется в виде:

{<c>, <T>}.

Например, если имеется отношение $\mathbf{T_1}$ с атрибутами ($\mathbf{a_1, a_2,..a_m}$) и отношение $\mathbf{T_2}$ с атрибутами ($\mathbf{b_1, b_2,..b_k}$), то команда

SELECT T1.A1, T1.A3, T2.B2, T2.B4, T2.B5 FROM T1, T2

будет представлена в виде:

$$<1><1,2>\{<1,3>,<2,4,5>\}, \qquad (2)$$

а команда

DELETE FROM T1 WHERE T1.A3='*string*'

будет представлена в виде: <4> <1>.

Учитывая, что вектора **<T>,** и **<A>** в (1) имеют одинаковую размерность, их совокупность представляется в виде произведения $\mathbf{cTA^T}$. Тогда выражения (2) принимает вид:

$$<1,1,1,3>,<1,2,2,4,5> \tag{3}$$

Для регистрации параметров можно использовать как встроенную в СУБД систему трассировки, таки и использовать систему триггеров, настроенных на определенные действия над таблицами (кроме команды Select).

V Построение решающего правила (классификатора)

Для обнаружения аномалий в поведении пользователей будем использовать так называемый *наивный байесовский классификатор* [6].

В основе классификатора лежит условная вероятность появления события $\boldsymbol{C}$ зависящего от нескольких переменных $\boldsymbol{F_1 \dots F_n}$.

$$p(C \mid F_1,....,F_n) \tag{4}$$

В нашем случае под событием $\boldsymbol{C}$ будем понимать факт принадлежности пользователя к роли $\boldsymbol{R_i}$,который определяется (предсказывается) на основании мониторинга текущей активности пользователя $\boldsymbol{F_1 \dots F_n}$. Проблема заключается в том, что когда количество свойств **n** очень велико или когда каждая из переменных $\boldsymbol{F_i}$ может принимать большое количество значений (что характерно для больших баз данных), тогда затраты на построение модели классификатора становятся неприемлемо большими. Поэтому преобразуем выражение (4).

Используя теорему Байеса, запишем

$$p(C \mid F_1,....,F_n) = \frac{p(C)\,p(F_1,...,F_n \mid C)}{p(F_1,...,F_n)} \tag{5}$$

В дальнейшем знаменатель выражения (5) не рассматриваем, поскольку он не зависит от $\boldsymbol{C}$ и при заданных $\boldsymbol{F_1 \dots F_n}$ является константой.

Числитель выражения (5) эквивалентен совместной вероятности

$$p(C, F_1,..., F_n) \tag{6}$$

которая, в предположении условной независимости свойств $\boldsymbol{F_i}$, может быть переписана в виде [6]:

$$\begin{aligned} &p(C, F_1,...., F_n) = p(C)\,p(F_1 \mid C)\,p(F_2 \mid C)\,p(F_3 \mid C)... = \\ &p(C)\prod_{i=1}^{n} p(F_i \mid C) \end{aligned} \tag{7}$$

Это означает, что из предположения о независимости параметров $\boldsymbol{F_1 \dots F_n}$, условное распределение переменной $\boldsymbol{C}$ может быть представлено в виде:

$$p(C \mid F_1,....,F_n) = \frac{1}{Z}\,p(C)\prod_{i=1}^{n} p(F_i \mid C) \tag{8}$$

где $\boldsymbol{Z}$ — это масштабный множитель, зависящий только от $\boldsymbol{F_1 \dots F_n}$, то есть константа, если значения переменных известны.

Оценка параметров модели (8) – это задача построение профиля для ролей $\boldsymbol{R_1 \dots R_m}$ по данным регистрации.

Решающее правило строится на основе принципа максимума апостериорной вероятности, которое еще называют апостериорным правилом принятия решения (MAP). Роль, отвечающая принципу максимума апостериорной вероятности, определяется как:

$$R_{MAP} = \arg\max_{R_j \in R} p(R_j) \prod_{i=1}^{n} p(F_i \mid R_j) \quad (9)$$

В соответствии с выбранным подходом каждая составляющая выражения (3), заключенная в угловые скобки считается независимым событием. Относительные частоты появления таких событий для роли R_j являются оценками условных вероятностей $p(F_i|R_j)$. В случае, если проверяется транзакция, состоящая из нескольких команд, верхний предел n в выражении (9) равен количеству независимых событий вида (3), составляющих транзакцию.

Принцип обнаружения аномального поведения следующий – если на основании модели и текущего поведения пользователя классификатор определяет, что пользователь принадлежит к роли R_j и он действительно включен в эту роль, то поведение пользователя считается нормальным, в противном случае оно считается аномальным.

Литература: **1.** *Carter and Katz. Computer crime: an emerging challenge for law enforcement. FBI Law Enforcement Bulletin, 1-8, December 1996.* **2.** *Y. Hu and B. Panda. Identification of malicious transactions in database systems. In Proceedings of the nternational Database Engineering and Applications Symposium (IDEAS), 2003.* **3.** *A. Kamra, E. Bertino, and E. Terzi. Detecting anomalous access patterns in relational databases. The International Journal on Very Large Data Bases (VLDB), 2008.* **4.** *C. Chung, M. Gertz, and K. Levitt. Demids: a misuse detection system for database systems. In Proceedings of Integrity and Internal Control in Information Systems: Strategic Views on the Need for Control. IFIP TC11 WG11.5 Third Working Conference, 2000.* **5.** *V. Lee, J. Stankovic, and S. Son. Intrusion detection in realtime databases via time signatures. In Proceedings of the Sixth IEEE Real-Time Technology and Applications Symposium (RTAS), 2000.* **6.** *T. M. Mitchell. Machine Learning. McGraw-Hill, 1997.*

УДК 004.056.5+003.26

АНАЛИЗ СТОЙКОСТИ МЕТОДА КОХА-ЖАО СТЕГАНОГРАФИЧЕСКОГО ВСТРАИВАНИЯ ИНФОРМАЦИИ В СТАТИЧЕСКИЕ ИЗОБРАЖЕНИЯ

Дмитрий Андрущенко, Галина Козина
Запорожский национальный технический университет

Аннотация: **Рассмотрен метод стеганографического встраивания информации Коха-Жао. В статье проведен анализ стойкости данного метода к JPEG-компрессии изображений со встроенным сообщением. Разработаны рекомендации по выбору параметров алгоритма.**
Summary: **The steganographic Koch and Zhao method is considered. The robustness of this method to the JPEG-compression of images with embedded data is analyzed. The robust algorithm settings are recommended.**
Ключевые слова: **Стеганоанализ, статическое изображение, метод Коха-Жао, алгоритм сжатия JPEG.**

I Введение

В связи с широким распространением мультимедийных технологий в последние годы появился значительный интерес к стеганографии. За это время было опубликовано немало качественных алгоритмов стеганографического скрытия данных в изображениях, как в зарубежной, так и отечественной литературе [1–4]. Однако значительно меньше публикаций посвящено анализу стойкости предложенных алгоритмов к различным атакам. Стеганографических методов, одинаково стойких ко всем видам атак, на сегодняшний день не существует. Поэтому при выборе стеганоалгоритма важно иметь в наличии как можно более подробный анализ стойкости этих алгоритмов к различным видам атак.

Другим важным требованием к стеганосистемам является «незаметность» встроенного сообщения, для обеспечения которого искажения, вносимые в контейнер во время скрытия в нем информации, должны быть минимальными, но обеспечивать при этом необходимую стойкость к определенным видам атак. В данной работе исследована стойкость стеганографического метода Коха-Жао к атаке сжатия JPEG в зависимости от различных параметров реализации алгоритма [1].